AD 274 198



Aeronutronic Publication No. C-1569
Air Force Document No. SSD-TDR-62-22

RESEARCH LABORATORIES

# TECHNICAL REPORT

## PERFORMANCE OPTIMIZATION OF MULTI-COMPONENT PROPELLANT MIXTURES

FINAL REPORT

VOLUME I OF II

MATHEMATICAL PROCEDURES AND PROGRAMMING

Prepared for: Air Force Flight Test Center
Air Force Systems Command
United States Air Force
Edwards Air Force Base, California

Under Contract: AF 04(611)-7020

Project Manager: S. E. Stephanou

Work done by: B. Kubert, D. Piper, L. Sashkin, R. Baier, R. Oliver, S Stephanou

12 January 1962

AERONUTRONIC
A DIVISION OF Ford Motor Company
FORD ROAD / NEWPORT BEACH, CALIFORNIA

## ABSTRACT

A computer program is described which can directly determine the composition which yields maximum impulse for multicomponent propellant mixtures.

The method has been coded for IBM 709 and 7090 computers and has been demonstrated for systems containing up to four components. The mathematics have been determined so that the technique is applicable to systems containing up to ten components but, thus far, it has only been applied to systems containing two to five components.

The computation proceeds directly to the optimum point; consequently, an economy of machine time over conventional procedures is realized. The program can be used in conjunction with any accurate performance computational program.

The final report is made up of two volumes: Volume I describes the mathematical development and procedures adopted for carrying out the optimization process and shows computer program flow charts; Volume II presents the results obtained when the program was applied to some multi-component systems of current interest and discusses some interesting aspects of impulse surfaces.

CONTENTS

VOLUME I - MATHEMATICAL PROCEDURES AND PROGRAMMING

CONTENTS (Continued)

# ILLUSTRATIONS

# SECTION 1

# INTRODUCTION

A computer program has been developed which can directly determine rapidly and economically the propellant composition which yields maximum impulse. The method has been coded for IBM 709 and IBM 7090 computers and has been demonstrated for systems containing up to four components. The mathematics have been determined so that the technique is applicable to systems containing up to ten components, but such complex multicomponent systems have not been tested.

In the past, when binary systems have been considered, the procedure for finding the composition yielding maximum impulse has consisted of selecting weight ratios of the two ingredients such that certain major product species would be formed. Computations of performance were made of a number of such mixture ratios and a curve drawn through the points to determine the maximum. Usually, five or more such computations, depending upon the accuracy desired, yielded the composition of maximum impulse. If really exact determination of the theoretical maximum was desired, more calculations were needed, particularly if the performance curve was irregular.

For three-component systems, the most convenient and reliable representation is the use of triangular diagrams. As an alternative to triangular plots, Cartesian coordinates can also be used, but these can lead to errors and omissions and are not as clear cut. Using the triangular plot technique, a number of compositions are again arbitrarily chosen and plotted. When sufficient numbers of impulse values are determined, at least 12 in the best possible cases and usually many more for the average system, constant impulse contour lines are estimated and drawn in the plot. Again the accuracy of the final result is a function of the number of computations made. A great many more points are usually required for ternary systems than for binary systems, inasmuch as it is

difficult to guess where the optimum-performing composition will be. For four or more component systems, the number of calculations required becomes inordinately large, and representation of the results and attainment of the true maximum becomes increasingly difficult.

The computer program described in this report minimizes the number of performance calculations required to determine the composition of maximum impulse. Valuable computer time is saved by using the previously calculated composition, or some close approximation to it, as a starting composition in each step of the optimization process. In addition to economizing machine time and directly determining the optimum composition, the program has been set up to allow restriction of the ingredients to certain ranges of values or ratios, as desired. Thus, for a solid propellant system where the presence of some minimum amount of binder is necessary, even though it degrades impulse, the binder content is not allowed to drop below a certain prespecified value. Similarly, the ratio of oxidizer to fuel can be maintained while the binder content can be varied, etc. These additions to the program take into account practical considerations which cannot be ignored even in theoretical work.

The mathematical details and the programing of the optimization procedure are described in Sections 2 through 7 of Volume I. Some additional programing to improve the efficiency of the program is suggested in Section 8 of Volume I. Volume II presents the results obtained when the program was applied to some multicomponent systems of current interest.

# SECTION 2

# MATHEMATICAL APPROACH TO THE PROBLEM

## 2.1 GENERAL CONSIDERATIONS

Due to the complexity of the relation between the reactants involved in a rocket motor performance calculation and the specific impulse produced by the combustion of these reactants, it was felt that the only fruitful approach to the optimization of specific impulse would be one of numerical nature. In the course of developing the current optimization program, two such approaches were attempted. These were:

(1) approximation of the impulse surface by a second order surface, and

(2) a gradient approach.

In each of these methods an initial composition of reactants is chosen, and by considering certain properties of the specific impulse function in the neighborhood of this initial point, a new point is determined. Hopefully, the specific impulse at this new point will be greater than that at the old. Let us denote the old and new points by y and z, respectively, and the specific impulse at some arbitrary point x by $I(x)$. Further, we will assume that if y is not the point with maximum impulse, then the application of our process to y will produce a z such that $I(z) > I(y)$. Clearly, by using z as a new y the process can be repeated with perhaps a further increase in $I(z)$. This iteration can be continued until no further improvement is possible.

Note that this procedure embodies the assumption that there is only one relative maximum of $I(x)$ in the domain of x where x is a closed set. For functions $I(x)$ that do not satisfy this condition the procedure will determine some point, say $x^*$, at which a relative maximum of $I(x)$ occurs; however, $I(x^*)$ will not necessarily be the absolute maximum of $I(x)$.

## 2.2 SELECTION OF INDEPENDENT VARIABLES

The point x must represent the reactants involved in a particular specific impulse calculation. For instance, the $i^{th}$ coordinate of x (i.e., $x_i$) could be the mass, or perhaps the number of moles of the $i^{th}$ reactant. With either of these definitions of x, however, we would have that $I(x) = I(kx)$ where k is any positive constant. This follows, since I(x) is a specific quantity, i.e., independent of total amount, and x and kx represent mixtures of reactants of the same relative amounts. This situation, that is,where an infinity of points represents a single composition, has obvious computational disadvantages. The basic problem is, that for mixtures of n reactants, specific impulse is a function of only n-1 independent variables. For instance, we could choose the ratios of the amounts of the first n-1 reactants to that of the last one as the n-1 coordinates of x. Another possibility would be to represent the amounts of each of the n reactants as a coordinate of x but require that the total amount, that is $\sum_{i=1}^{n} x_i$, be constant. If this constant is one, then the $x_i$ become mass or mole fractions.

SECTION 3

# OPTIMIZATION USING A QUADRATIC APPROXIMATION

The initial approach to the optimization problem used a quadratic approximation to I(x) at y. This attempt was not successful, and therefore will be treated only briefly.

Let $x_i$ represent the mass fraction of the $i^{th}$ reactant in the mixture. The quadratic approximation to I(x) given as a truncated Taylor Series about the point y is

$$Q(x) = I(y)+\sum_{i=1}^{n} \left.\frac{\partial I}{\partial x_i}\right|_{x=y} \Delta_i + \sum_{i=1}^{n}\sum_{j=1}^{n} \left.\frac{\partial^2 I}{\partial x_i \partial x_j}\right|_{x=y} \Delta_i \Delta_j \qquad (1)$$

where $\Delta_i = x_i - y_i$.

Since the $x_i$ are mass fractions, we have

$$\sum_{i=1}^{n} x_i = 1 \qquad (2)$$

We will take the new point z to be that point which causes Q(x) to be an extremum subject to the constraint given by equation (2). Applying the Lagrange Multiplier technique, we are led to the following equations:

$$\lambda + \sum_{i=1}^{n} \left.\frac{\partial^2 I}{\partial x_i \partial x_j}\right|_{x=y} \Delta_i = -\left.\frac{\partial I}{\partial x_j}\right|_{x=y} \qquad j=1,2,\ldots n \qquad (3)$$

where $\lambda$ is a Lagrange Multiplier. Equation (2) must be satisfied for both the old and new points. It follows that

$$\sum_{i=1}^{n} \Delta_i = 0 \qquad (4)$$

The partial derivatives in equations (3) must, of course, be evaluated numerically. Equations (3) and (4) are a set of n+1 linear equations in the unknowns $\lambda$ and the $\Delta_i$. The solution to this system gives corrections which, when applied to the point y, give the new point z. Thus

$$z_i = y_i + \Delta_i \qquad i=1,2,\ldots n \qquad (5)$$

The algorithm just described makes the tacit assumption that I(x) can be represented reasonably well by a second order surface in n dimensional space. For many systems, however, this is not the case. This can readily be surmised from an examination of the projection of the impulse surface in various planes. This is discussed in more detail in the section titled "Characteristics of Some Impulse Surfaces".

Due to the poor approximation obtained by using second order surfaces, the relation $I(z) > I(y)$ would not in general hold. Therefore, this approach to the problem was abandoned.

SECTION 4

# OPTIMIZATION USING THE GRADIENT APPROACH

The method that was actually used in the computer program is a modification of one described by Curry[1]. Other modifications of this procedure have been proposed in the literature. Box's method[2] seems to be an excellent one; however, it is not easily mechanized for machine computation. The method used by Booth for the solution of linear equations[3] is clearly not adequate for the optimization of systems whose specific impulse function is extremely rugged. The method described herein is easily mechanized and seems to be sufficiently powerful to handle up to four component systems with no difficulty. The following discussion includes the option of imposing linear constraints on the reactants, although this option has not been included in the computer program. With the imposition of m linear constraints, the procedure should easily handle systems of 4+m components.

Let $x_i$ be the mass fraction of $i^{th}$ reactant of the system*. As before, we have the restriction that

$$\sum_{i=1}^{n} x_i = 1 \tag{6}$$

Now suppose that we impose m linear constraints on the $x_i$, to wit:

$$\sum_{i=1}^{n} a_{ik} x_i = b_k \qquad k=1,2,\ldots m \tag{7}$$

---

*An optimization program was also written using mass fraction ratios as the components of x. This procedure was somewhat inferior to the method that was adopted for the existing program.

Note that equation (6) is of the same form as equation (7). We shall include them altogether by writing

$$\sum_{i=1}^{n} a_{ik}x_i = b_k \qquad k=0,1,2,\ldots m \tag{8}$$

where $b_0$ and all the $a_{i0}$ have the value 1.

Let us solve equation (8) for $x_i, i=1,2\ldots m+1$ in terms of the remaining $x_i$. Thus

$$\begin{bmatrix} a_{10} & a_{20} & \cdots & a_{m+1,0} \\ a_{11} & a_{21} & \cdots & a_{m+1,1} \\ \vdots & \vdots & & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{m+1,m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m+1} \end{bmatrix} = \begin{bmatrix} b_0 - \sum_{i=m+2}^{n} a_{i0}x_i \\ b_1 - \sum_{i=m+2}^{n} a_{i1}x_i \\ \vdots \\ b_m - \sum_{i=m+2}^{n} a_{im}x_i \end{bmatrix} \tag{9}$$

Let A be the determinant of coefficients in equation (9) and let $A_i$ be A with its $i^{th}$ column replaced by the column vector on the right side of (9). Then, by Cramer's Rule

$$x_i = \frac{A_i}{A} \qquad i=1,2,\ldots m+1 \tag{10}$$

where A is a non-zero constant and the $A_i$ are functions of $x_i$, $i=m+2,m+3\ldots n$ only. (The requirement that $A\neq 0$ is equivalent to requiring the constraints given by equation (8) to be independent.) It follows that specific impulse, $I_{sp}$, is a function of only n-m-1 of the reactants; that is

$$I_{sp} = I(x_{m+2}, x_{m+3}, \ldots x_n) \tag{11}$$

We will denote the quantity n-m-1 as the number of degrees of freedom of the system.

In addition to the constraints given by equation (8), we must require that the mass fraction, $x_i$, of each reactant be non-negative. It is no more difficult to require that $x_i$ be not less than an arbitrary positive constant, $e_i$, consistent with equation (8). For the sake of generality, we shall follow this course. Thus, for the n-m-1 independent variables, we have

$$x_i \geq e_i \qquad i=m+2,m+3...n \qquad (12)$$

and for the remaining variables from equation (10)

$$A_i \geq Ae_i \qquad i=1,2,...m+1 \qquad (13)$$

We summarize the results thus far obtained as follows: We wish to find the numbers $x_i$, $i=m+2,m+3,...n$ subject to the inequalities (12) and (13) that will cause $I_{sp}$, given by the expression (11), to assume a maximum value. (Note that once the $x_i$ for $i=m+2,m+3,...n$ are fixed, the remaining $x_i$ can be obtained from equation (9).

The procedure at this point as described in Reference 1 would be to start at some point $y=(y_{m+2},y_{m+3},...y_n)$ and proceed in the direction of steepest ascent, i.e., in the direction of grad $I_{sp}$, to a point z such that $I(z) > I(x)$ for all points x on the line through y with direction grad $I_{sp}$. At this time, the point z would be treated as new y and the process repeated until convergence was attained. Such a process is illustrated in Figure 1 for a system with two degrees of freedom. The dotted line composed of straight segments represents the path taken by the iteration. The curved lines represent level curves of $I_{sp}$. It is easily shown that any two adjacent segments of the path are such that the first is tangent to a level curve of $I_{sp}$ at the point of intersection of the two segments while the second is normal to the same curve at the same point.

Figure 1 illustrates a surface whose maximum value is easily found by applying the algorithm just described. Unfortunately, the situation is not always this simple; for example, an examination of Figure 2 would indicate that very many steps would be necessary in order to locate the peak for that system. It would seem then that we should not be restricted to travel only in the direction of grad $I_{sp}$.

Consider a typical step in an optimization path for a system with two degrees of freedom, as shown in Figure 3. The points y' and y represent the previous and current reactant compositions, respectively. V' is a unit vector in the direction of the step from y' to y, and U is a unit vector in the direction of grad $I_{sp}$ at y. By hypothesis, there is one and only one relative maximum of $I_{sp}$ in dom x; suppose it occurs at x*. Denote the unit vector in the direction from y to x* by V. It is clear that if $I_{sp}$ is continuous in dom x*, then $U \cdot V > 0$; that is, U and V

---

*dom x indicates the region in which x is defined.

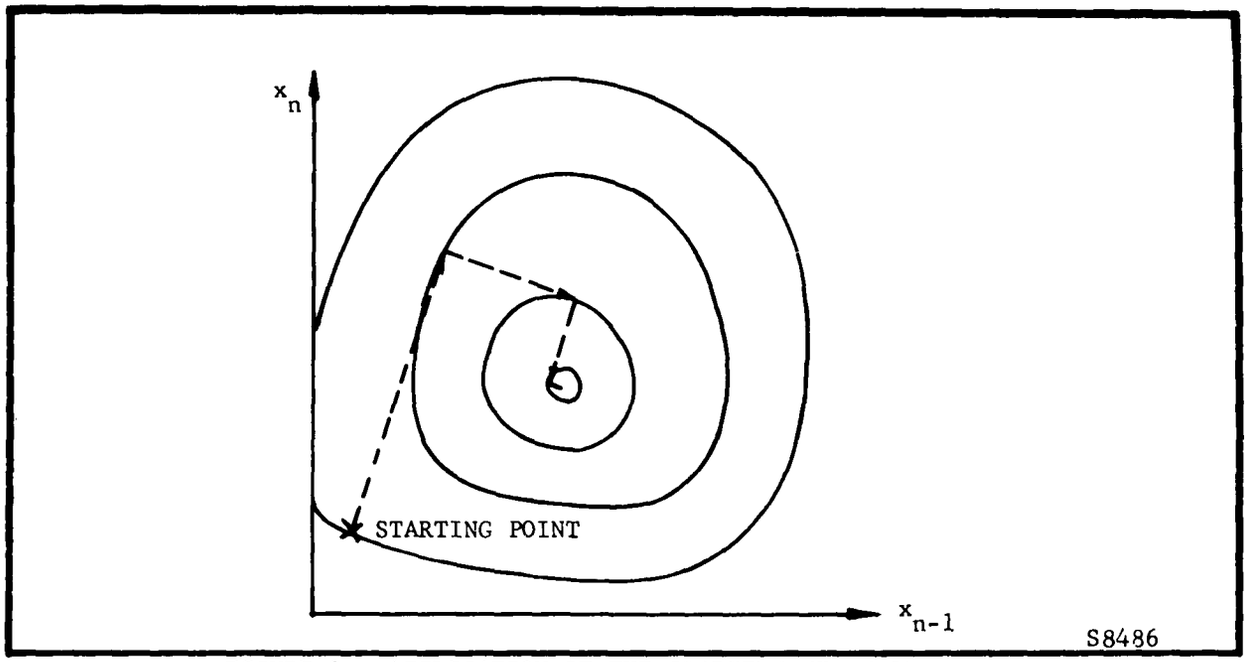



FIGURE 1. APPLICATION OF THE GRADIENT TECHNIQUE TO A SIMPLE SYSTEM WITH TWO DEGREES OF FREEDOM

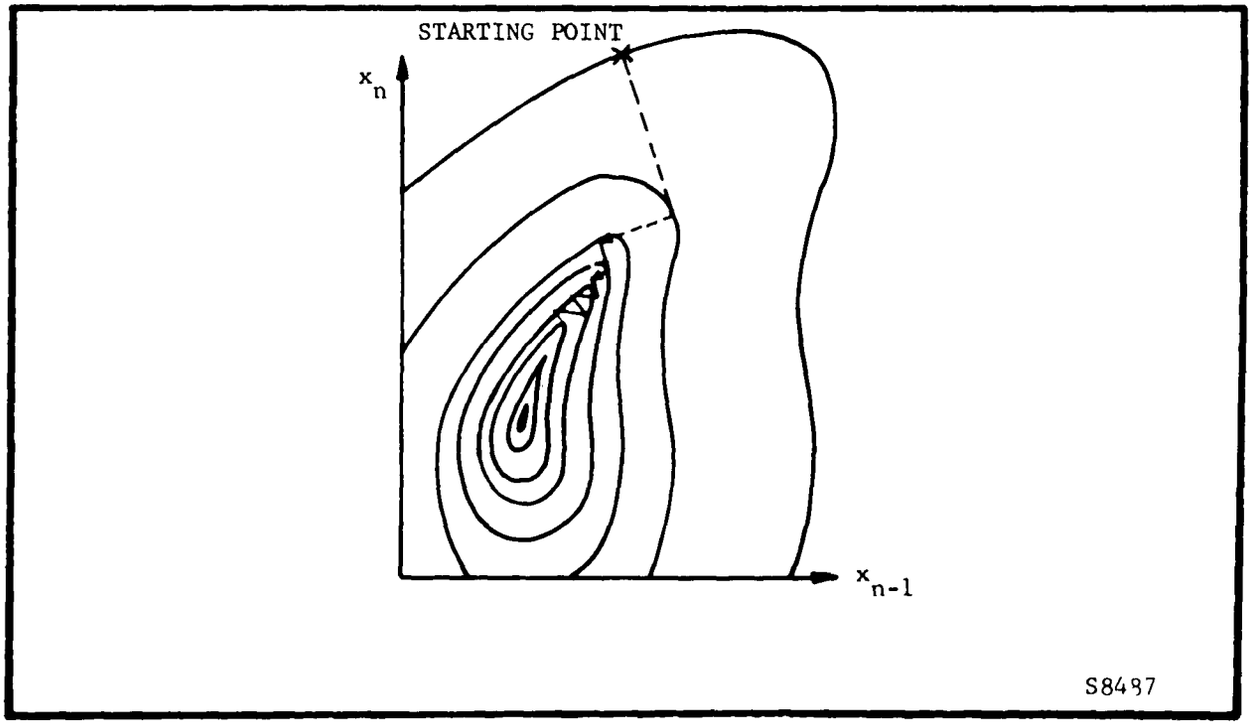



FIGURE 2. APPLICATION OF THE GRADIENT TECHNIQUE TO A LESS REGULAR SYSTEM THAN THAT SHOWN IN FIGURE 1.

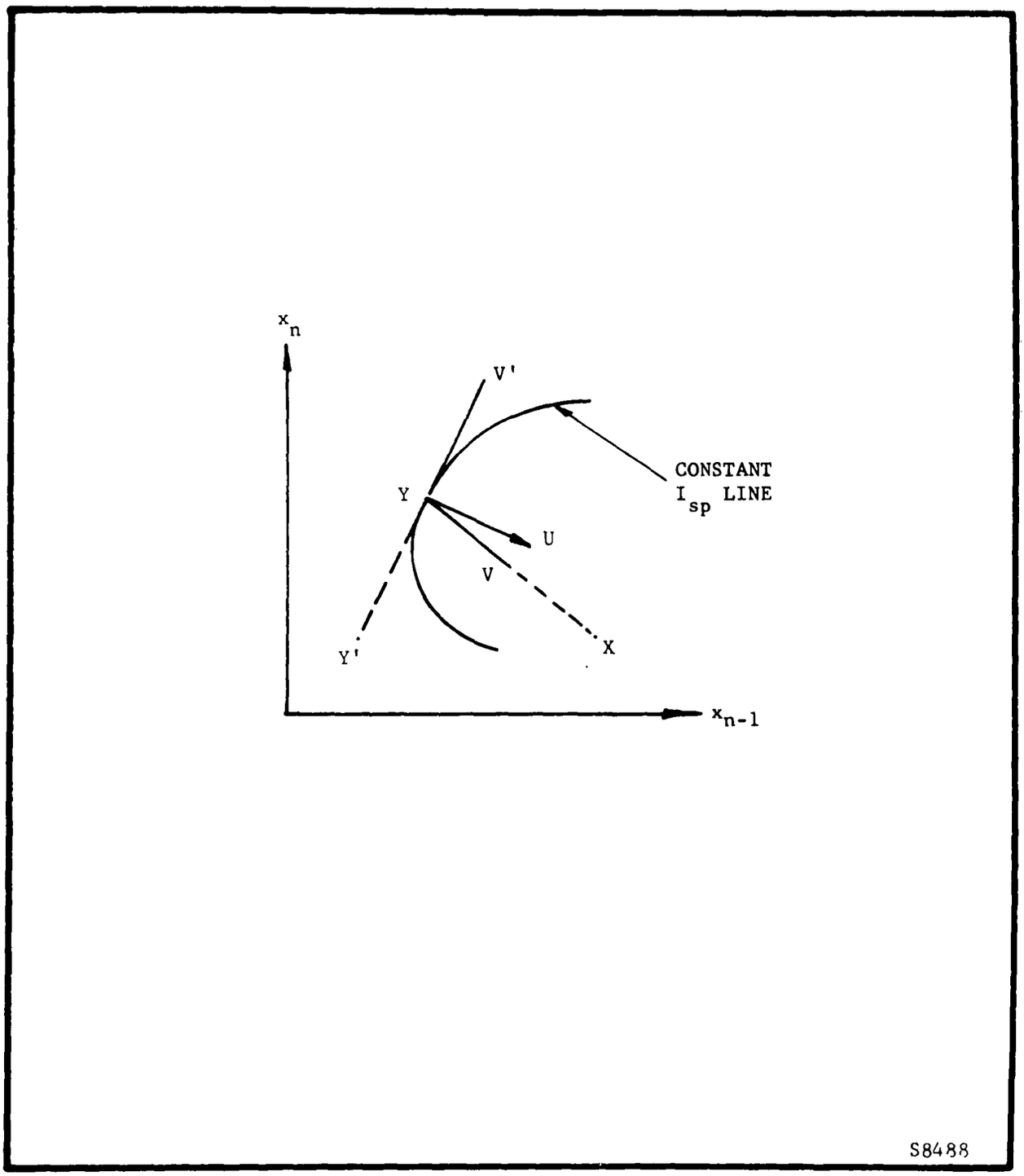



FIGURE 3. MODIFICATION OF THE GRADIENT TECHNIQUE USING THE "T PROCEDURE"

will both lie on the same side of V'. It follows that we can represent V in the form

$$V = \sqrt{1-t^2}\,U + tV' \qquad -1<t<1 \qquad (14)$$

For systems with more than two degrees of freedom, the point x* will in general not be the optimum point for the system, but will be the optimum point in the plane defined by U and V'.

Suppose we allow the variable t in equation (14) to take on any value between -1 and 1. Then any point x, in the half plane on the side of V' toward U, is given by

$$x = y + \lambda V = y + \lambda\left[\sqrt{1-t^2}\,U + tV'\right] \qquad (15)$$

where $\lambda$ is the distance of x from y and U has components

$$u_i = \frac{\partial I/\partial x_i}{\left[\sum_{j=m+2}^{n}\left(\frac{\partial I}{\partial x_i}\right)^2\right]^{\frac{1}{2}}} \qquad i=m+2, m+3, \ldots n \qquad (16)$$

Since U, V' and y are fixed at any point in the iteration, x is a function of only $\lambda$ and t. We emphasize this relation by rewriting equation (11) as follows:

$$I_{sp} = I(\lambda, t) \qquad (17)$$

For a specified value of t, $I_{sp}$ is a function of $\lambda$ only. Denote this function as $I_t(\lambda)$. Let $\lambda^*$ be the value of $\lambda$ that maximizes $I_t(\lambda)$. Clearly, $\lambda^*$ will depend on t. Let us restrict the $\lambda$ in equation (17) to take on only the values $\lambda^*$. Then

$$I_{sp} = I(\lambda^*(t), t) = I^*(t) \qquad (18)$$

that is, we can consider $I_{sp}$ as a function of t only. The point at which $I^*(t)$ is a maximum is clearly $x^*$.

The functions $I_t(\lambda)$ and $I^*(t)$ can be maximized by successive polynomial approximations. This is further discussed in the following section.

The domain of x, (i.e., the points representing acceptable reactant compositions) is defined by the inequalities (12) and (13). Consider

the equations formed from (12) and (13) by replacing the inequalities by equalities. Thus,

$$x_i = e_i \qquad i=m+2,m+3,\ldots n \qquad (19)$$

$$A_i = Ae_i \qquad i=1,2,\ldots m+1 \qquad (20)$$

Equations (19) and (20) represent planes in the n-m-1 dimensional x space. Dom x is the region contained within all the planes; that is, if y is some point in dom x, then any other point x in dom x can be joined to y by a line segment that does not cross any of the planes given by equations (19) and (20).

The above considerations make it clear that $\lambda$ in equation (15) may be limited to certain values, say $0 < \lambda < \lambda_{max}$ where $\lambda_{max}$ is the value that would cause x to lie on the closest plane, given by equations (19) and (20), in the direction of V. Let us denote the components of V by $v_i, i=m+2,m+3,\ldots n$. A line through y in the direction of V is given by

$$\frac{x_i - y_i}{v_i} = \lambda \qquad i=m+2,m+3,\ldots n \qquad (21)$$

Now $\lambda$ is the distance between x and y. Denote the distance, along the direction of V, between y and the $j^{th}$ limiting plane given by equations (19) or (20) as $\lambda_j$. Then, from equation (19)

$$\lambda_j = \frac{e_j - y_j}{v_j} \qquad j=m+2,m+3,\ldots n \qquad (22)$$

In order to find $\lambda_j$, for the remaining j, substitute $x_i$ given by equation (21) into equation (20). Thus, for $j=1,2,\ldots m+1$

$$A_j = \begin{vmatrix} a_{10} & a_{20} \cdots b_0 - \sum_{i=m+2}^{n} a_{i0}(y_i + \lambda_j v_i) \cdots a_{m+1,0} \\ a_{11} & a_{21} \cdots b_1 - \sum_{i=m+2}^{n} a_{i1}(y_i + \lambda_j v_i) \cdots a_{m+1,1} \\ \vdots & \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ a_{1m} & a_{2m} \cdots b_m - \sum_{i=m+2}^{n} a_{im}(y_i + \lambda_j v_i) \cdots a_{m+1,m} \end{vmatrix} = Ae_j \qquad (23)$$

where the distinct column is column j. Define

$$B_j = \begin{vmatrix} a_{10} & a_{20} \cdots b_0 - \sum_{i=m+2}^{n} a_{i0}y_i \cdots a_{m+1,0} \\ a_{11} & a_{21} \cdots b_1 - \sum_{i=m+2}^{n} a_{i1}y_i \cdots a_{m+1,1} \\ \vdots & \vdots \qquad\qquad \vdots \qquad\quad \vdots \\ a_{1m} & a_{2m} \cdots b_m - \sum_{i=m+2}^{n} a_{im}y_i \cdots a_{m+1,m} \end{vmatrix} \tag{24}$$

and

$$C_j = \begin{vmatrix} a_{10} & a_{20} \cdots \sum_{i=m+2}^{n} a_{i0}v_i \cdots a_{m+1,0} \\ a_{11} & a_{21} \cdots \sum_{i=m+2}^{n} a_{i1}v_i \cdots a_{m+1,1} \\ \vdots & \vdots \qquad\qquad \vdots \qquad\quad \vdots \\ a_{1m} & a_{2m} \cdots \sum_{i=m+2}^{n} a_{im}v_i \cdots a_{m+1,m} \end{vmatrix} \tag{25}$$

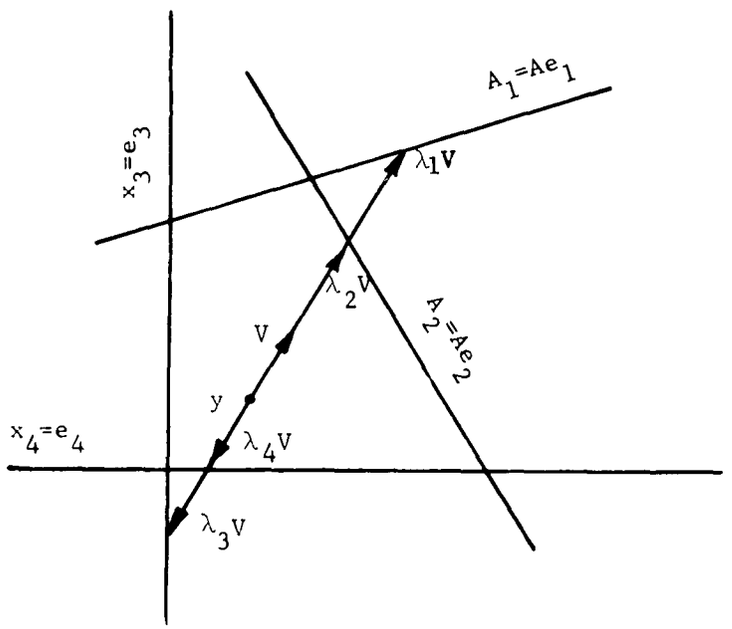
where the distinct column is column j. Then equation (23) can be written

$$B_j - \lambda_j C_j = Ae_j$$

so that

$$\lambda_j = \frac{B_j - Ae_j}{C_j} \qquad j=1,2,\ldots m+1 \tag{26}$$

Equations (22) and (26) give expressions for the distance, along the direction of V, between y and the $j^{th}$ limiting plane. The situation is illustrated in Figure 4. Here we have taken n=4 and m=1. Dom x is the closed finite region bounded by the lines $A_1=Ae_1, A_2=Ae_2, x_3=e_3$ and $x_4=e_4$. In this case V points toward the former two lines and away from the latter two. Thus here $\lambda_1, \lambda_2 > 0$ and $\lambda_3, \lambda_4 < 0$. It is clear that $\lambda_2$ is the largest step that can be taken, in the direction of V, starting from y, if the point x is to remain in dom x. In general, the largest step is given by




S8499

FIGURE 4. LIMITING PLANE RESTRICTIONS ON MAXIMUM STEP SIZE IN DOM $\chi$

$$\lambda_{max} = \text{min pos } \gamma_j \qquad j=1,2,\ldots n \qquad (27)$$

where min pos stands for "the minimum positive value of" and

$$\gamma_j = \frac{B_j - Ae_j}{C_j} \qquad j=1,2,\ldots m+1$$

$$\gamma_j = \frac{e_j - y_j}{v_j} \qquad j=m+2, m+3 \ldots n$$

## SECTION 5

## OPTIMIZATION ON A LINE

The discussion thus far has provided the basis for choosing a region R, that is a finite portion of some half plane in x-space, in which an improved composition is to be sought. The actual seeking out of this improved point is done by maximizing $I_{sp}$ in R. In order to accomplish this, it is necessary to be able to locate the optimum point along some line L in R. The development of an algorithm that will efficiently locate the optimum point on L necessitates the assumption that one and only one relative maximum of $I_{sp}$ exists on L. This assumption does not follow from our original hypothesis; i.e., that one and only one relative maximum of $I_{sp}$ exists in dom x, and, in fact, is not generally valid. For example, see Figure 5. Here, there is a unique maximum in dom x; however, if we restrict x to points on L, then the resulting function has two relative maxima*. None of the cases that have thus far been run on the computer have been affected by the invalidity of this assumption. Since the assumption enables a powerful tool to be utilized in the optimization process, it will be made; however, it must be kept in mind that this procedure could provide a possible source of difficulty in the optimization of a given system. As will be noted below, this assumption can, in effect, be removed from the optimization process by the suitable choice of a program constant. This, of course, should not be done unless trouble arises in using the recommended procedure.

---

*The difficulty arises in the maximization of the function I*(t) given in equation (18). The numerical procedure used in computing I*(t) is such that it is possible for either of the maxima indicated in Figure 5 to be specified as the optimum point on L. The resulting function I*(t) could therefore be discontinuous.

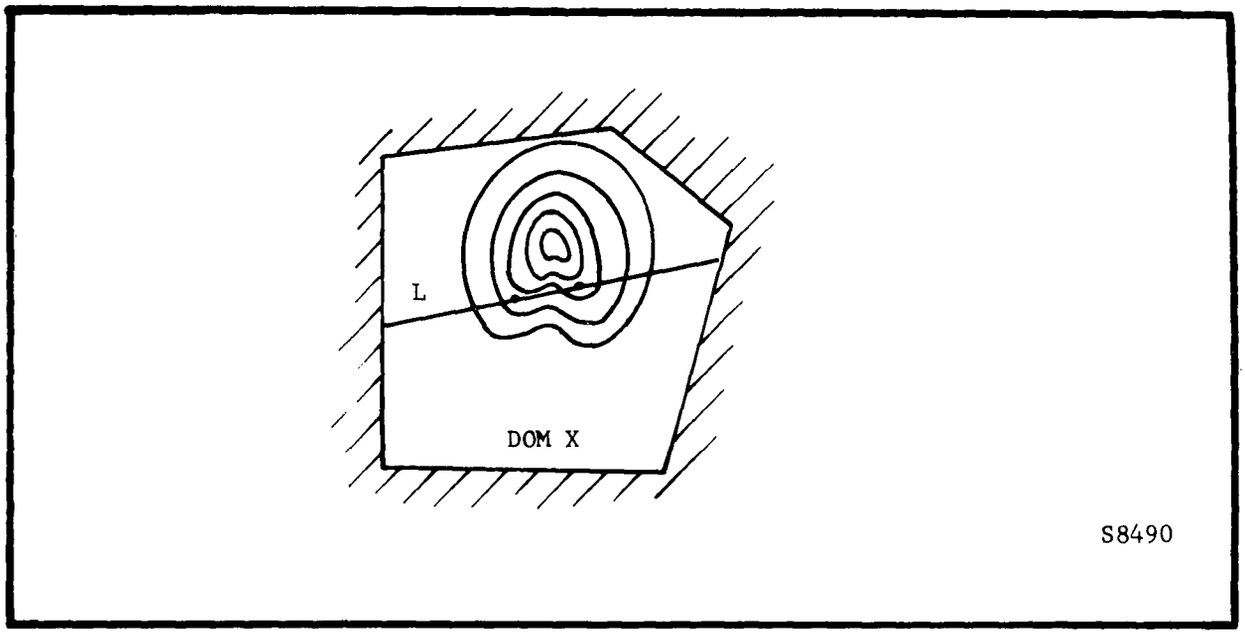



FIGURE 5. EXAMPLE OF TWO RELATIVE MAXIMA ALONG A LINE

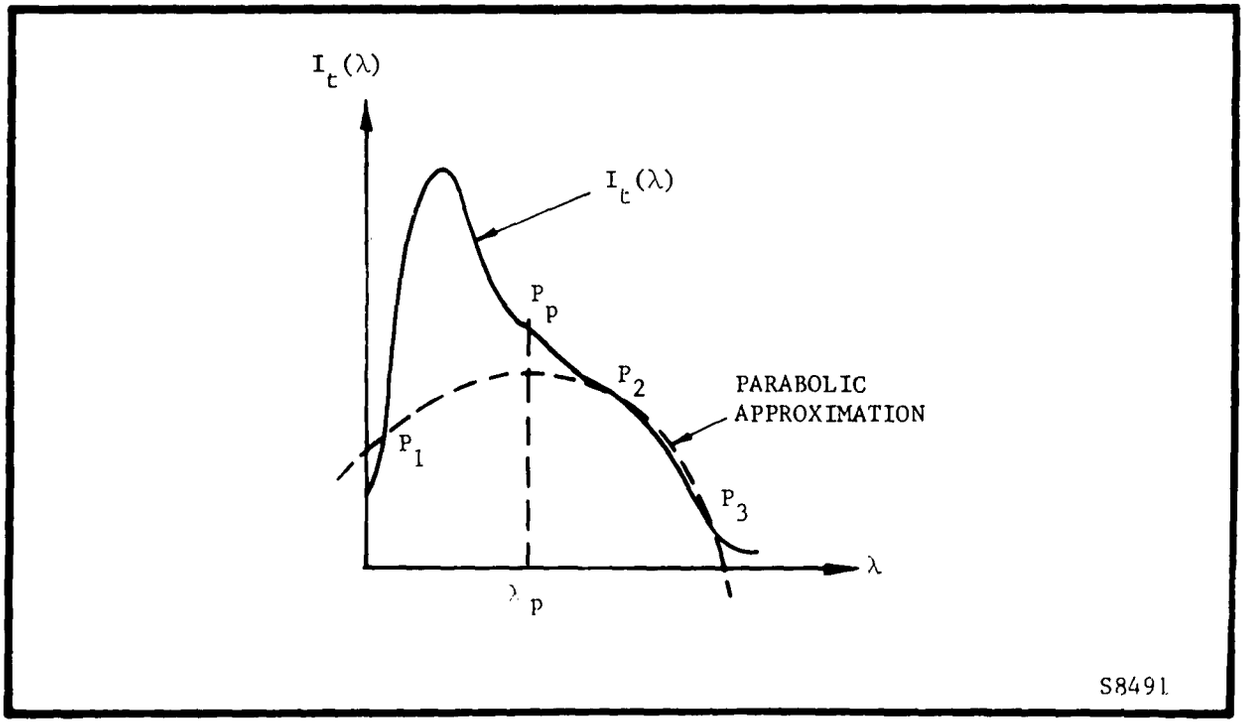



FIGURE 6. TECHNIQUE FOR LOCATING MAXIMUM LYING WITHIN DOM X

Consider the function defined by equation (17) for a specified value of t. The region of definition for this function is L, the line segment composed of the points x given by equation (15) for $0 \leq \lambda \leq \lambda_{max}$ where $\lambda_{max}$ is given by equation (27). We have already referred to this function as $I_t(\lambda)$. It will be assumed that one and only one relative maximum of $I_t(\lambda)$ exists. Under this assumption, there are three possible situations:

(a) $I_t(0)$ is the relative maximum,

(b) $I_t(\lambda_{max})$ is the relative maximum, or

(c) a relative maximum exists for $0 < \lambda < \lambda_{max}$.

The process of locating the optimum $\lambda$ offers no problems in the first two cases. Accordingly, we will examine the third case.

Consider a graphical representation of $I_t(\lambda)$ in which the ordinate and abscissa of a point $P_i$ represents $I_t(\lambda_i)$ and $\lambda_i$, respectively. Clearly, we can determine three values of $\lambda$, say $\lambda_1 < \lambda_2 < \lambda_3$, such that $I_t(\lambda_2) \geq I_t(\lambda_i)$, i=1,3. Through the corresponding points $P_1$, $P_2$ and $P_3$, let us construct a parabola, and then determine the value of $\lambda$ corresponding to the vertex of the parabola, say $\lambda_p$. Next, we can compute $I_t(\lambda_p)$ and thereby obtain $P_p=(\lambda_p, I_t(\lambda_p))$. From the four points $P_1$, $P_2$, $P_3$ and $P_p$, we can choose that with the largest ordinate. (This, of course, will be either $P_2$ or $P_p$.) Using this as a new $P_2$, together with the adjacent points on either side of it as new $P_1$ and $P_3$, we obtain a new set of three points, and the process can be repeated. This is illustrated in Figure 6. Here, the next set $P_1$, $P_2$, $P_3$ would be the old points $P_1$, $P_p$, $P_2$.

The program actually employs a process that is a slight modification of the procedure just described. The trouble with the above procedure is that it can be rather slow in converging. For example, consider Figure 7. The initial three points are A, B and C. In this case the curve is such that a poor approximation to the optimum $\lambda$ is obtained at each step. The process yields, successively, the points D, E, F, G and H, and, in turn, discards, successively, the points B, C, D, E and F. Here, of course, the troublesome point is A, and its retention prevents good approximation to the optimum

In order to alleviate this difficulty, the current procedure sometimes determines, in addition to $P_p$, the point $P_m=(\lambda_m, I(\lambda_m))$ where $\lambda_m=\frac{1}{2}(\lambda_1+\lambda_3)$. This is done only if $\lambda_m$ is considerably different than both $\lambda_2$ and $\lambda_p$. The criterion being used is that if both

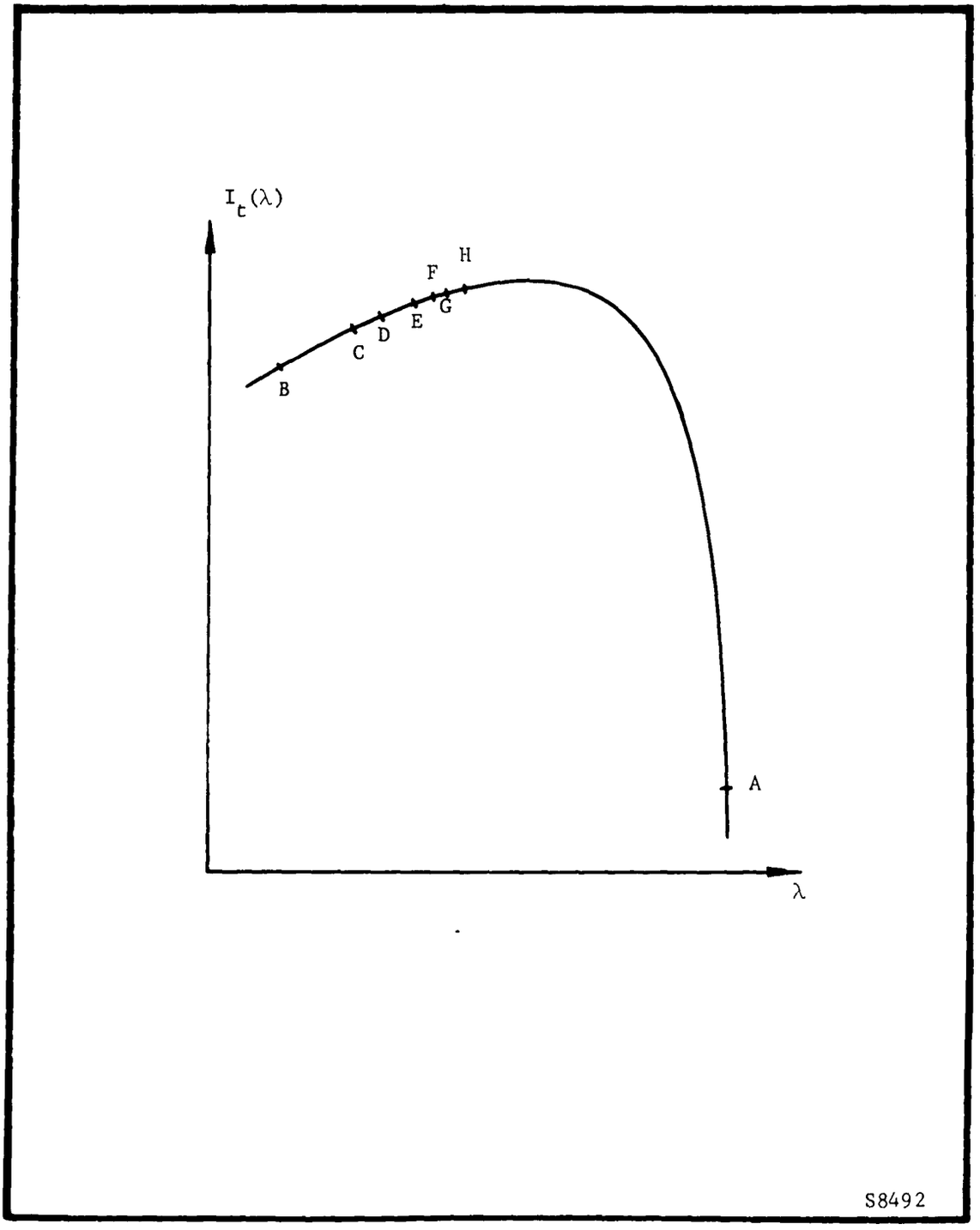



FIGURE 7. EXAMPLE OF SLOW CONVERGENCE TO MAXIMUM

$\frac{\lambda_m - \lambda_2}{\lambda_3 - \lambda_1} \geq 0.2$ and $\frac{\lambda_m - \lambda_p}{\lambda_3 - \lambda_1} \geq 0.2$ then $P_m$ is determined. The actual process is as follows:

1. Start with a set of points $(P_1, P_2, P_3)$.
2. Compute $\lambda_m$ and $\lambda_p$.
3. If $\frac{\lambda_m - \lambda_p}{\lambda_3 - \lambda_1} < 0.2$, go to 8.
4. If $\frac{\lambda_m - \lambda_2}{\lambda_3 - \lambda_1} < 0.2$, go to 8.
5. Otherwise, compute $I_t(\lambda_m)$, thereby obtaining $P_m$.
6. Get new set $(P_1, P_2, P_3)$ from old points $(P_1, P_2, P_3, P_m)$ as described above.
7. Compute $\lambda_p$ from new set.
8. Compute $I_t(\lambda_p)$ thereby obtaining $P_p$.
9. Get new set $(P_1, P_2, P_3)$ from old points $(P_1, P_2, P_3, P_p)$ as described above.
10. Go to 2.

Whenever a new set of points is obtained, they are tested in order to determine if convergence has been achieved and therefore if the iteration should be terminated.

The foregoing discussion gives the method used in evaluating $I^*(t)$ defined in equation (18). Once t is specified, the optimum $\lambda$, i.e., $\lambda^*(t)$, can be determined by the above procedure. Thus $I^*(t)$ is obtained. In finding the value of t that will maximize $I^*(t)$, a similar procedure is used. In this case the ten steps outlined above are repeated except that all $I_t(\lambda)$ are replaced with $I^*(t)$ and all $\lambda$ are replaced with t.

The determination of the maximum value of I*(t) is a rather lengthy process when compared to the computation involved in the maximization of $I_t(\lambda)$. In order to avoid unnecessary computation and still retain the use of the I*(t) maximization when applicable, the following procedure is effected.

(a) Grad $I_{sp}$ is determined, and $I_o(\lambda)$ is maximized. This computation provides $\lambda^*(0)$ as defined in conjunction with equation (18).

(b) If $\lambda^*(0) > e_\lambda$ where $e_\lambda$ is a program constant (presently, we are using 0.02), then go to (a).

(c) Otherwise maximize I*(t).

(d) Go to (a).

Note that if $e_\lambda$ is taken to be zero, then the maximization of I*(t) is completely bypassed, and the process reduces to the usual gradient method as described by Curry, in Reference 1. In any case, where the assumption of a unique maximum of $I_{sp}$ on a line causes computational difficulties, this procedure can be followed.

SECTION 6

DETERMINATION OF THE GRADIENT

Let $P = (X_{m+2}, X_{m+3}, \ldots X_n)$ be an arbitrary point in the n-m-1 dimensional x-space. The impulse function, $I_{sp}$, can be considered as a surface, I(P), in the n-m dimensional space. If $P_o = (Y_{m+2}, Y_{m+3}, \ldots Y_n)$ is some specified point in the x-space, then the tangient hyperplane to the impulse surface at $P_o$ is given by

$$I(P) - I(P_o) - \sum_{i=m+2}^{n} \left.\frac{\partial I}{\partial X_i}\right|_{P_o} (X_i - Y_i) = 0 \qquad (28)$$

The derivatives in equation (28) are the components of grad $I_{sp}$ at $P_o$. Since they are not directly obtainable we must use equation (28) in order to evaluate them numerically. Thus we can choose points $P_j$, $j = m+2, m+3, \ldots n$ in the neighborhood of $P_o$ and, after evaluating the $I(P_j)$, equation (28) yields n-m-1 linear equations in the unknowns $\left.\frac{\partial I}{\partial X_i}\right|_{P_o}$. Thus

$$I(P_j) - I(P_o) - \sum_{i=m+2}^{n} \left.\frac{\partial I}{\partial X_i}\right|_{P_o} (X_i^j - Y_i) \approx 0 \quad j = m+2, \ldots n \qquad (29)$$

where $X_i^j$ is the ith coordinate of $P_j$. The solution to equations (29) affords an approximation to grad $I_{sp}$ at $P_o$.

A simpler procedure would be to choose the $P_j$ such that only the $j^{th}$ coordinate of $P_j$ differs from that of $P_o$. In that case the system of equations (29) would reduce to

$$\left.\frac{\partial I}{\partial X_i}\right|_{P_o} \approx \frac{I(P_i) - I(P_o)}{X_i - Y_i} \qquad i = m+2, m\_3, \ldots n \qquad (30)$$

If there is some a priori knowledge of the nature of the impulse function it may be feasible to choose the $P_j$ in such a way as to minimize the error inherent in the approximate equations (29). However, in general we do not have such knowledge. Therefore, we shall prefer the simpler equation (30).

Let us define the point $\Delta P_i$ as follows

$$\Delta P_i = (0,0,0,\ldots \Delta X_i, 0, 0, \ldots 0) \qquad (31)$$

where $\Delta X_i = X_i - Y_i$ is the $i^{th}$ coordinate of the right side of equation (31). Then equation (30) becomes

$$\left.\frac{\partial I}{\partial X_i}\right|_{P_o} \approx \frac{I(P_o + \Delta P_i) - I(P_o)}{\Delta X_i} \qquad i = m+2, \ldots n \qquad (32)$$

Equations (32) are linear approximations to the derivatives at $P_o$.

It has been found that by improving the accuracy of the derivatives at each step the iteration to optimum $I_{sp}$ is accelerated. A more accurate set of derivatives can usually be obtained if, instead of using equations (32), a quadratic approximation is applied. If the points chosen for the quadratic approximation to $\left.\frac{\partial I}{\partial X_i}\right|_{P_o}$ are $P_o - \Delta P_i$, $P_o$, $P_o + \Delta P_i$ then the approximating form is

$$\left.\frac{\partial I}{\partial X_i}\right|_{P_o} \approx \frac{I(P_o + \Delta P_i) - I(P_o - \Delta P_i)}{2\,\Delta X_i} \qquad i = m+2, m+3, \ldots n \quad (33)$$

If equations (33) are used instead of equations (31), an additional impulse must be obtained for each i. The additional computation time, however, is small since $\Delta X_i$ is chosen to be a small number*; and so a good guess for the composition at $P_o - \Delta P_i$ is available. (The guess, of course, is the composition at $P_o$.) The program accordingly uses equations (32) to obtain the gradient.

* Currently we are using $\Delta X_i = \max (0.0015625\ X_i,\ 0.000625)$

SECTION 7

# PERFORMANCE PROGRAM REQUIREMENTS

The derivatives $\frac{\partial I}{\partial x_i}$ specified in equation (16) must be computed numerically. In order to determine these fairly accurately, rather small increments in $x_i$ must be utilized. This is especially true for systems with narrow ridges in the neighborhood of such ridges and for any system in the vicinity of its peak. This requirement of small $x_i$ increments necessitates a very precise specific impulse calculation.

Another facet of the optimization procedure that requires a precise evaluation of $I_{sp}$ is the one dimensional maximization of $I_t(\lambda)$ and also that of $I^*(t)$. In the vicinity of a one dimensional relative maximum it is likely that the approximating curve will be computed from points at close proximity. Thus a small deviation in the value of $I_{sp}$ at a point might radically change the nature of the approximating curve.

In computing numerical derivatives the computer program uses increments in $x_i$ of between 0.000625 and 0.0015. The program determines $I^*(t)$, i.e., the maximum value of $I_t(\lambda)$, to an accuracy of 0.001 in $\lambda$ and 0.001 in $I_{sp}$. These factors make it imperative that the performance program be precise (i.e., continuous) to within 0.0005 seconds in $I_{sp}$

The convergence criteria mentioned above could, of course, be somewhat relaxed. It has been noted, however, that in the rather prevalent case of a surface containing a narrow ridge, the determination of the peak impulse is accelerated by requiring rather precise convergence to be attained at intermediate points in the iteration. This is further discussed in the next section.

# SECTION 8

## SUGGESTED ADDITIONAL WORK

In order to make the program more useful and also more efficient, the following tasks should be undertaken:

(1) The option to impose linear constraints on the system should be programmed. The mathematical formulation of the inclusion of such constraints has been carried out as previously indicated.

(2) An improved method for finding the peak impulse along a line in mass fraction space should be sought. It would seem that a combination of cubic and parabolic approximations to the peak impulse would be preferable to the parabolic and mid-point approximation presently being used.

(3) If it is desired to optimize systems with four or more degrees of freedom, some consideration should be given to a generalization of the "t" iteration to higher dimensions.

(4) Special tests for convergence should be devised. In establishing convergence to the optimum propellant composition the current computer programs sometimes take an undue amount of time.

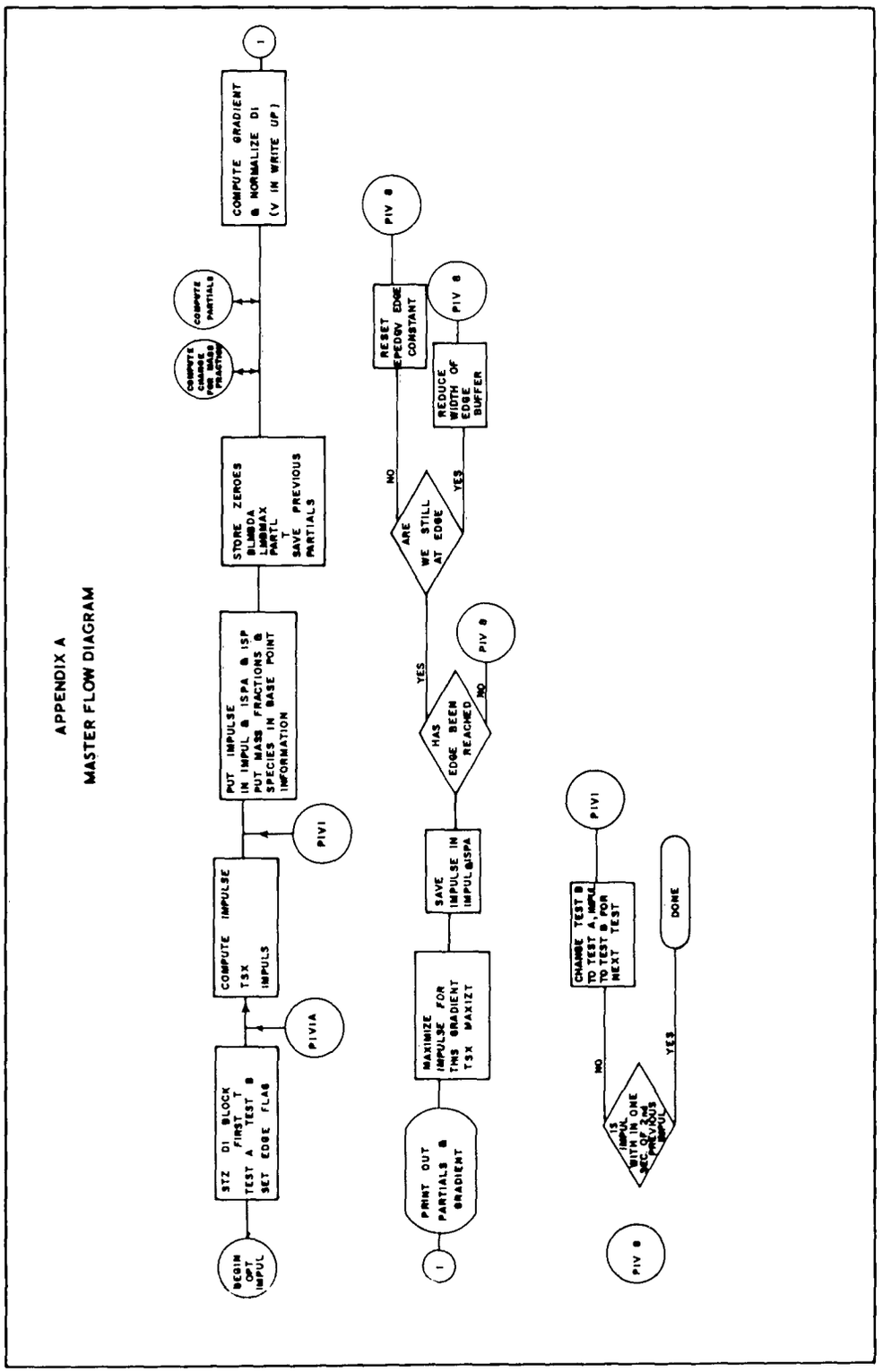
SECTION 9

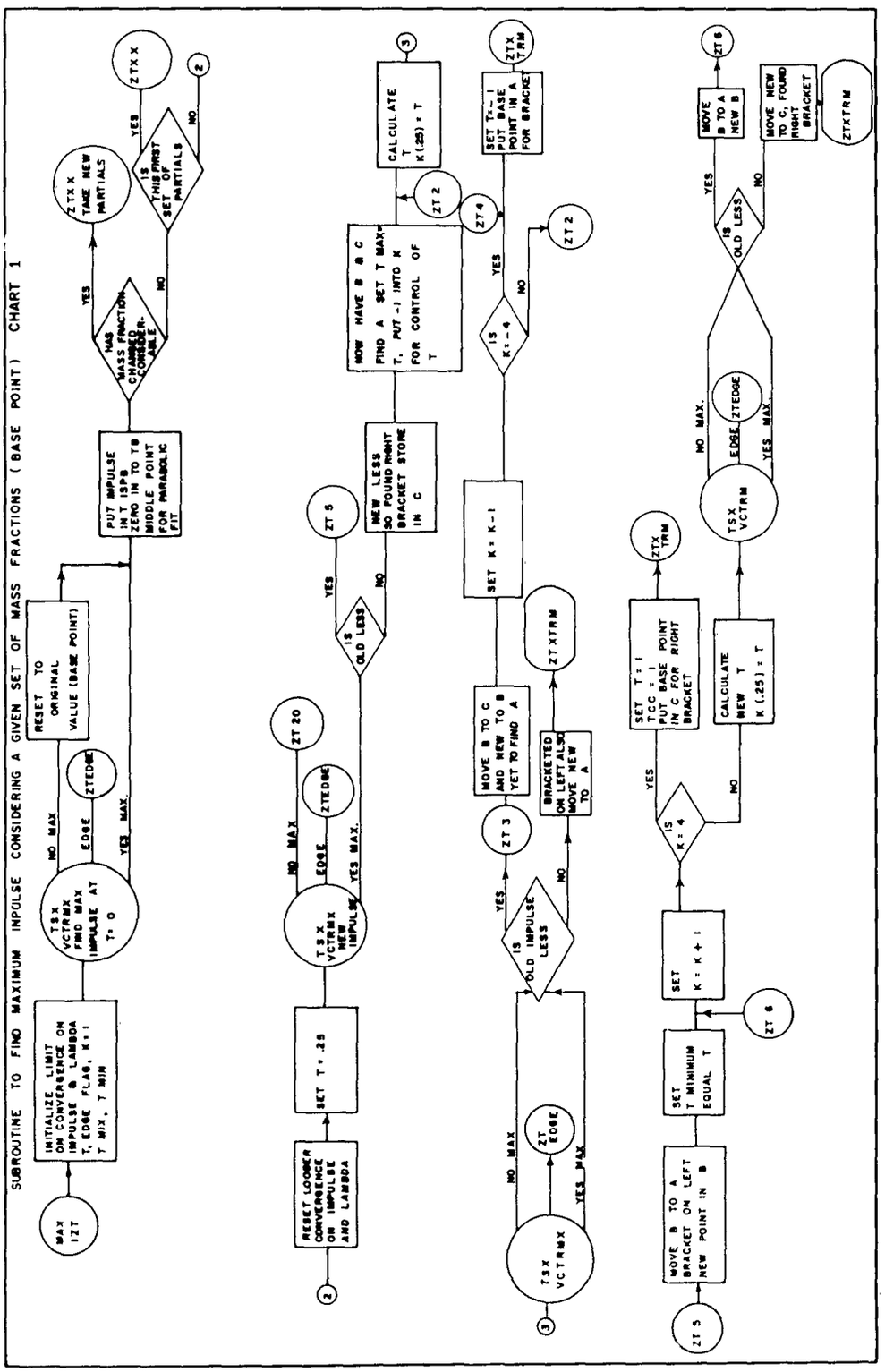
REFERENCES

1. Curry, H. B., The Method of Steepest Descent for Non-Linear Minimization Problems, Quart. Appl. Math. 2, (1944), 258-261.

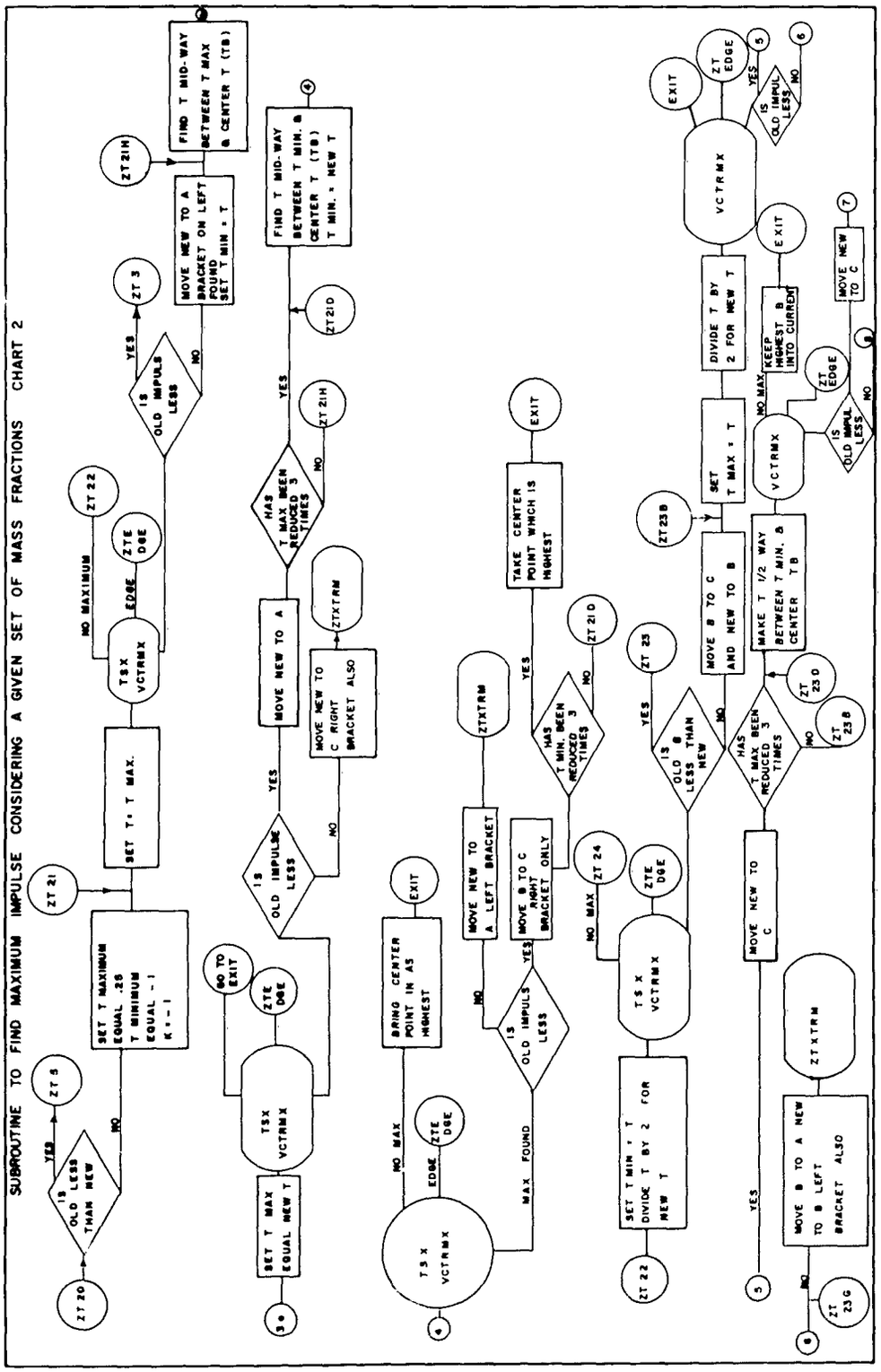
2. Davies, O. L. (Editor) (1956), The Design and Analysis of Industrial Experiments, Oliver and Boyd, pp. 495-578.

3. Booth, A. D. (1957), Numerical Methods, Butterworths, pp. 95-100.

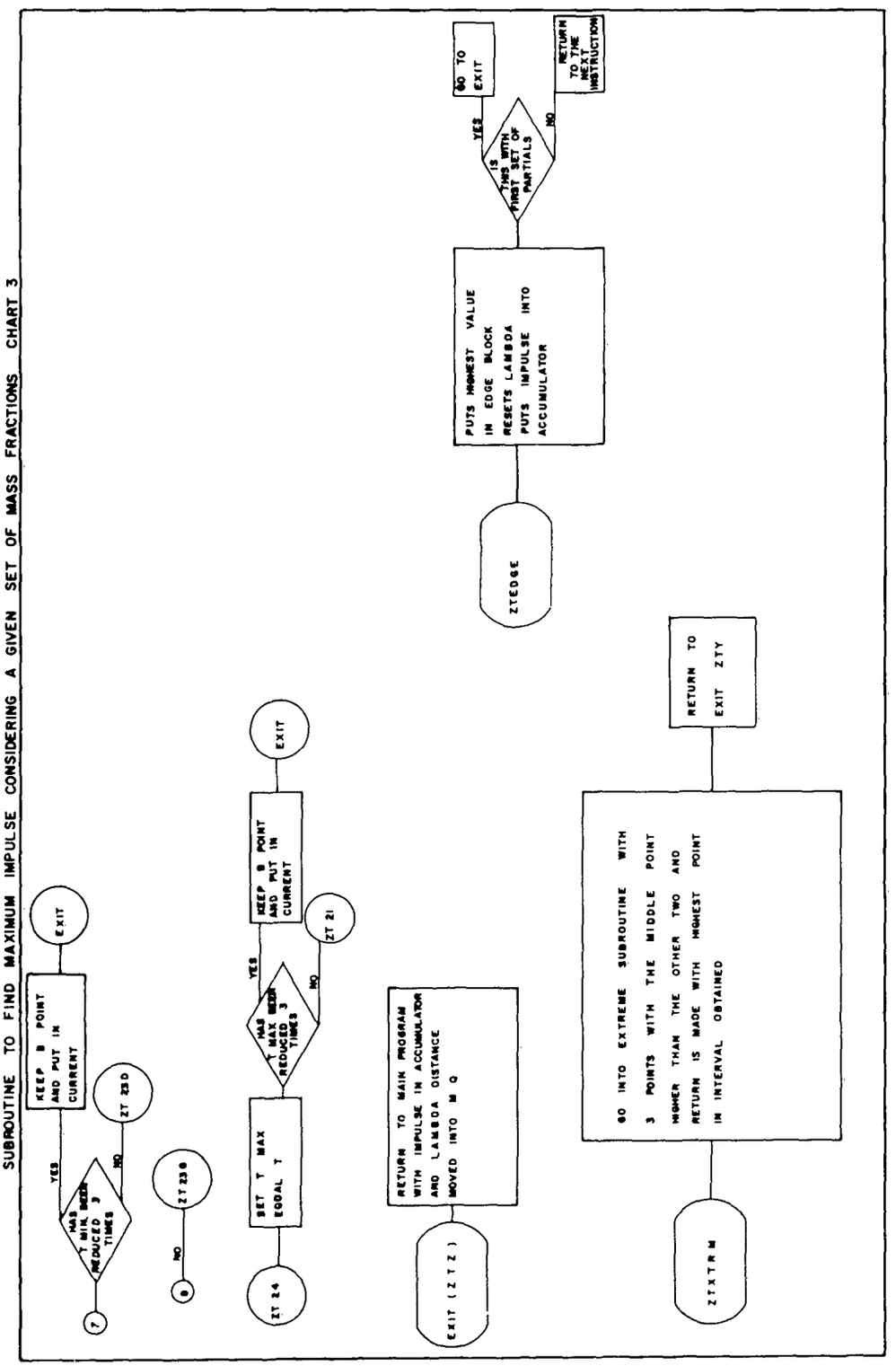

APPENDIX A
MASTER FLOW DIAGRAM
BEGIN OPT IMPUL
STZ DI BLOCK
FIRST T
TEST A TEST B
SET EDGE FLAG
PIVIA
COMPUTE IMPULSE
TSX
IMPULS
PIVI
PUT IMPULSE
IN IMPUL @ ISPA @ ISP
PUT MASS FRACTIONS @
SPECIES IN BASE POINT
INFORMATION
STORE ZEROES
BLMBDA
LMBMAX
PARTL
T
SAVE PREVIOUS
PARTIALS
COMPUTE PARTIALS
COMPUTE CHARGE PER MASS FRACTION
COMPUTE GRADIENT
& NORMALIZE DI
(V IN WRITE UP)
1
PRINT OUT
PARTIALS &
GRADIENT
1
MAXIMIZE
IMPULSE FOR
THIS GRADIENT
TSX MAXIZT
SAVE
IMPULSE IN
IMPUL@ISPA
HAS
EDGE BEEN
REACHED
YES
NO
PIV @
ARE
WE STILL
AT EDGE
NO
YES
RESET
EPEDGV EDGE
CONSTANT
REDUCE
WIDTH OF
EDGE
BUFFER
PIV @
PIV @
PIVI
CHANGE TEST B
TO TEST A, IMPUL
TO TEST B FOR
NEXT TEST
IS
IMPUL
WITHIN ONE
SEC OF 2nd
PREVIOUS
IMPUL
NO
YES
DONE
PIV @
AED-7101


APPENDIX B
SUBROUTINE TO FIND MAXIMUM INPULSE CONSIDERING A GIVEN SET OF MASS FRACTIONS (BASE POINT) CHART 1
MAX IZT
INITIALIZE LIMIT ON CONVERGENCE ON IMPULSE & LAMBDA T, EDGE FLAG, K=1 T MIX, T MIN
TSX VCTRMX FIND MAX IMPULSE AT T= 0
NO MAX
EDGE
ZTEDGE
YES MAX.
RESET TO ORIGINAL VALUE (BASE POINT)
PUT IMPULSE IN T ISPB ZERO IN TO TB MIDDLE POINT FOR PARABOLIC FIT
HAS MASS FRACTION CHANGED CONSIDER-ABLE
YES
NO
ZTXX TAKE NEW PARTIALS
IS THIS FIRST SET OF PARTIALS
YES
NO
ZTXX
2
RESET LOOSER CONVERGENCE ON IMPULSE AND LAMBDA
SET T = .25
TSX VCTRMX NEW IMPULSE
ZT 20
NO MAX
EDGE
ZTEDGE
YES MAX.
IS OLD LESS
YES
NO
ZT 5
NEW LESS SO FOUND POINT BRACKET STORE IN C
NOW HAVE B & C FIND A SET T MAX- T, PUT -1 INTO K FOR CONTROL OF T
CALCULATE T K(.25) = T
3
ZT 2
ZT 4
IS K = -4
YES
NO
ZT 2
SET T = 1 SET BASE POINT IN A FOR BRACKET
ZTX TRM
SET K = K-1
MOVE B TO C AND NEW TO B YET TO FIND A
ZT 3
IS OLD IMPULSE LESS
YES
NO
NO MAX
ZT EDGE
YES MAX
TSX VCTRMX
3
BRACKETED ON LEFT ALSO MOVE NEW TO A
ZTXTRM
ZTX TRM
SET T = 1 TSC = 1 PUT BASE POINT IN C FOR BRACKET
IS K = 4
YES
NO
CALCULATE NEW T K(.25) = T
TSX VCTRM
NO MAX.
EDGE
ZTEDGE
YES MAX.
IS OLD LESS
YES
NO
MOVE B TO A NEW B
ZT 6
MOVE NEW TO C, FOUND RIGHT BRACKET
ZTXTRM
SET K = K + 1
ZT 6
SET T MINIMUM EQUAL T
MOVE B TO A BRACKET ON LEFT NEW POINT IN B
ZT 5
AMD-7181

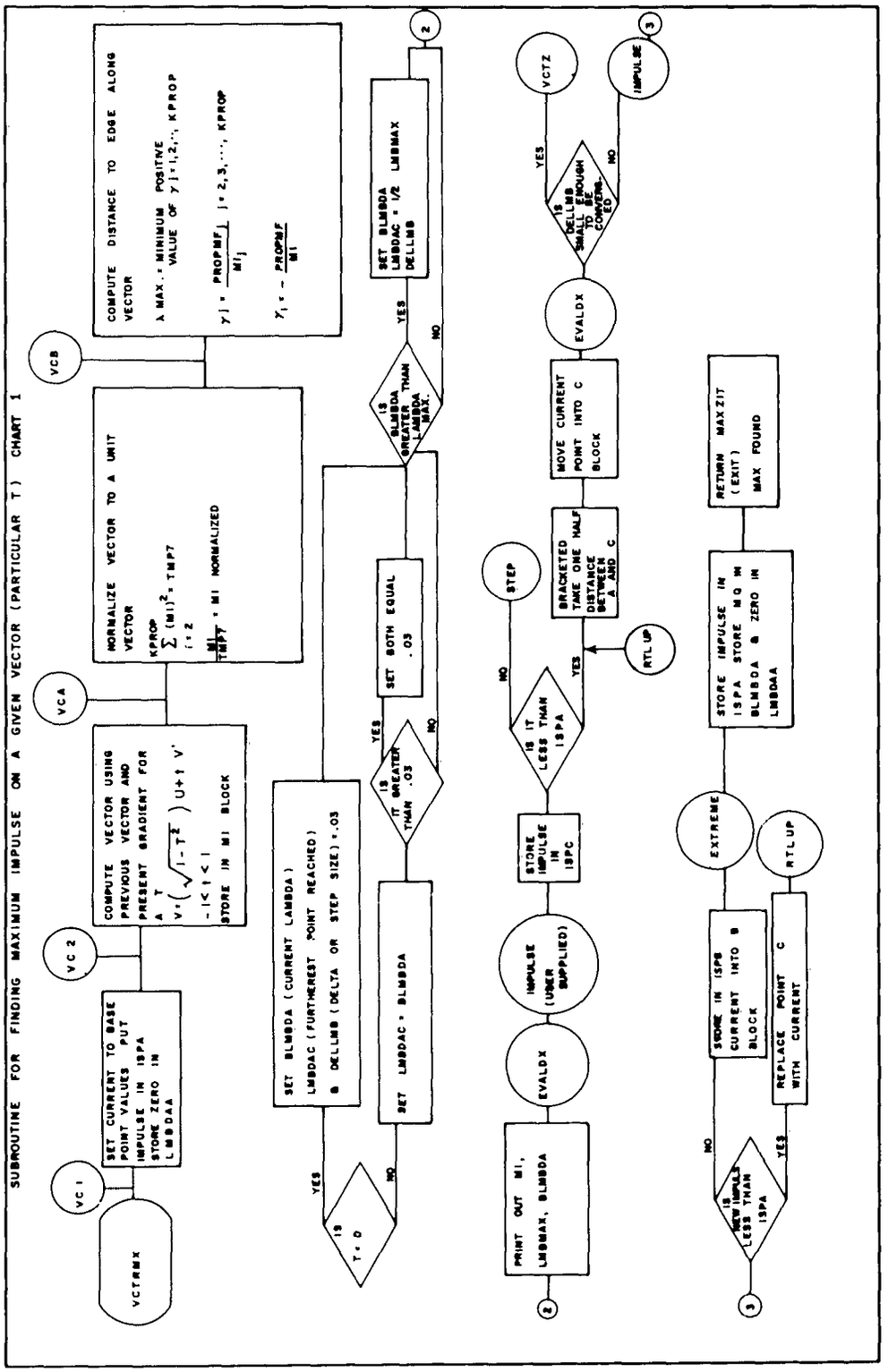
APPENDIX B (Continued)



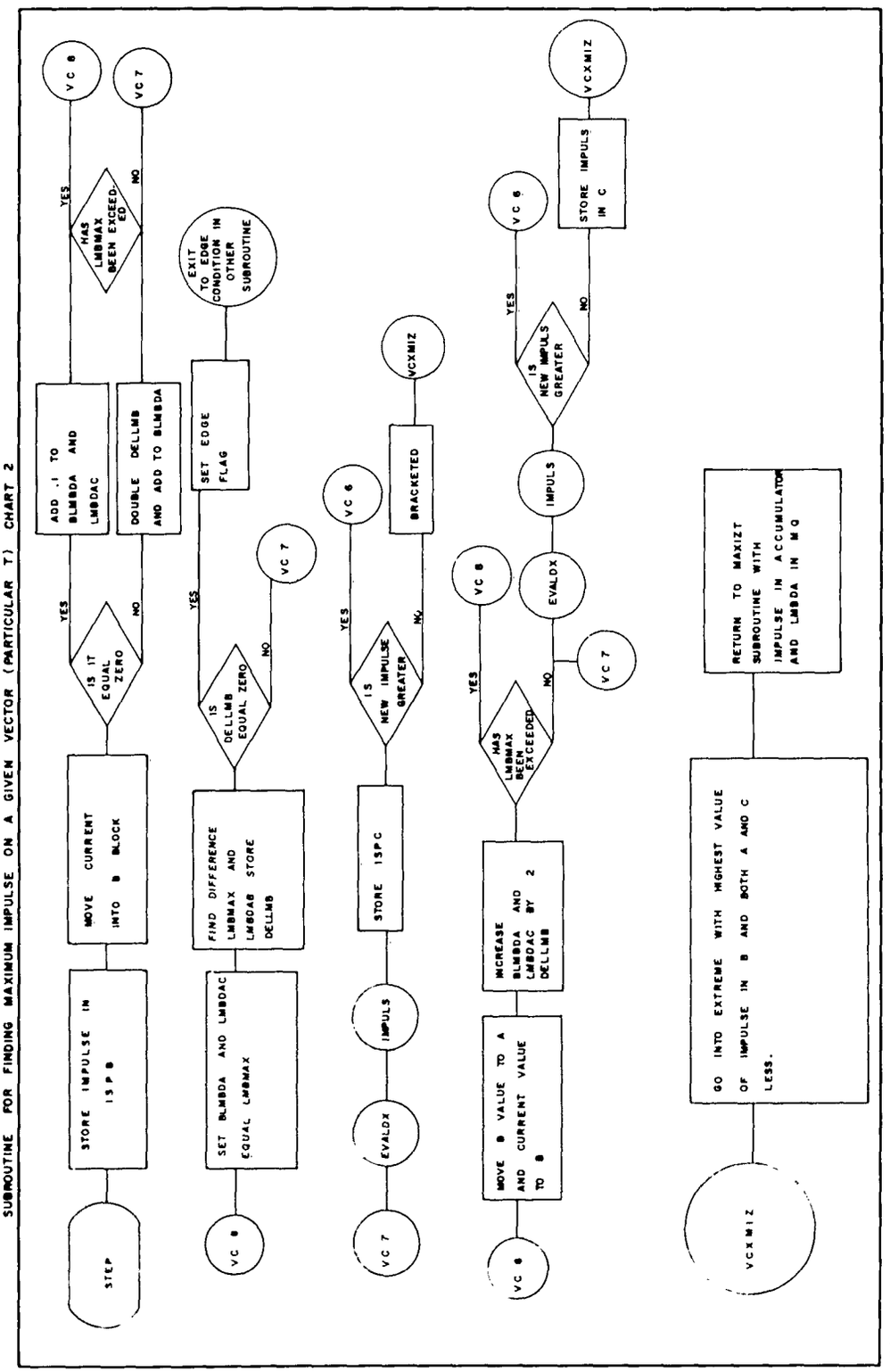
APPENDIX B (Continued)
SUBROUTINE TO FIND MAXIMUM IMPULSE CONSIDERING A GIVEN SET OF MASS FRACTIONS CHART 3



APPENDIX C


SUBROUTINE FOR FINDING MAXIMUM IMPULSE ON A GIVEN VECTOR (PARTICULAR T) CHART 1
VCTRMX
VC 1
SET CURRENT TO BASE POINT VALUES PUT IMPULSE IN ISPA STORE ZERO IN LMBDAA
VC 2
COMPUTE VECTOR USING PREVIOUS VECTOR AND PRESENT GRADIENT FOR A T
V' = (√(1 - T²)) U + T V'
-1 < T < 1
STORE IN MI BLOCK
VCA
NORMALIZE VECTOR TO A UNIT VECTOR
KPROP
Σ (MI)² = TMP7
I = 2
MI / TMP7 = MI NORMALIZED
VCB
COMPUTE DISTANCE TO EDGE ALONG VECTOR
A MAX. = MINIMUM POSITIVE VALUE OF γj j = 1,2,..., KPROP
γj = PROPMF_j / MIj  j = 2,3,..., KPROP
γ1 = - PROPMF / M1
SET BLMBDA LMBDAC = 1/2 LMBMAX DELLMB
IS BLMBDA GREATER THAN LAMBDA MAX.
YES
NO
2
SET BLMBDA (CURRENT LAMBDA) LMBDAC (FURTHEREST POINT REACHED) ± DELLMB (DELTA OR STEP SIZE) *.03
IS T = 0
YES
NO
SET LMBDAC = BLMBDA
IS IT GREATER THAN .03
YES
NO
SET BOTH EQUAL .03
PRINT OUT MI, LMBMAX, BLMBDA
EVALDX
IMPULSE (USER SUPPLIED)
STORE IMPULSE IN ISPC
IS IT LESS THAN ISPA
NO
YES
STEP
BRACKETED TAKE ONE HALF DISTANCE BETWEEN A AND C
RTLUP
MOVE CURRENT POINT INTO C BLOCK
EVALDX
IS DELLMB SMALL ENOUGH TO BE CONSIDERED
YES
NO
VCTZ
IMPULSE
3
EXTREME
STORE IN ISPB CURRENT INTO B BLOCK
IS IMPULSE LESS THAN ISPA
NO
YES
REPLACE POINT C WITH CURRENT
RTLUP
STORE IMPULSE IN ISPA STORE NO IN BLMBDA & ZERO IN LMBDAA
RETURN MAX Z1T (EXIT) MAX FOUND
A66-7101

APPENDIX C (Continued)
SUBROUTINE FOR FINDING MAXIMUM IMPULSE ON A GIVEN VECTOR (PARTICULAR T) CHART 2



APPENDIX D

SUBROUTINE FOR COMPUTING OPTIMUM VALUE OF A FUNCTION OF ONE VARIABLE

The following symbols are used in extrme charts 1 through 4:

XR - index register

FUNCT - The address of the starting location of the subroutine that computes the function to be extremized. NOTE: This subroutine must preserve index Register 1.

N - The maximum number of times that the function is to be computed.

TABLE - A block of storage in which all pertinent quantities are saved in the extremization process.

T - The level of the extremization (the optimization of I ( $\lambda$ ) is the 0-th level and the optimization of I*(t) is the 1st level.)

X - The starting location of a block of three words in which the abscissas of the bracketing points are to be found.

Y - Same as X except the ordinates are stored here.

ERRX - Address of cell containing tolerance for the abscissa

ERRY - Address of cell containing tolerance for the ordinate

P(x,y) - The point P having abscissa and ordinate x and y

m(subscript) - "At midpoint"

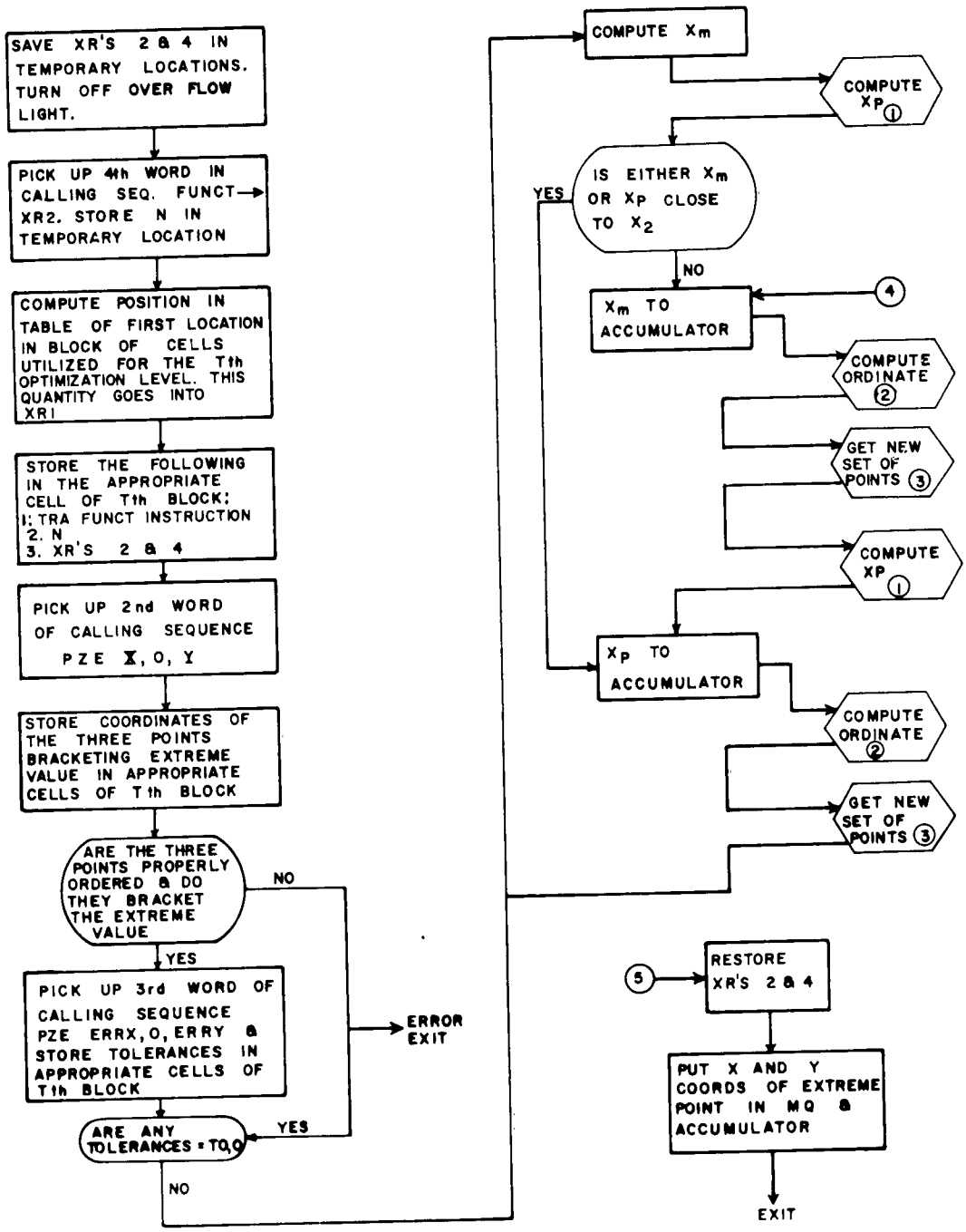
p(subscript) - "At parabolic approximation"

1, 2, and 3 (subscripts) - Leftmost, middle and rightmost points in the set of three bracketing points

AERONUTRONIC



## APPENDIX D (Continued)
## EXTRME CHART 1



## APPENDIX D (Continued)
## EXTRME CHART 2

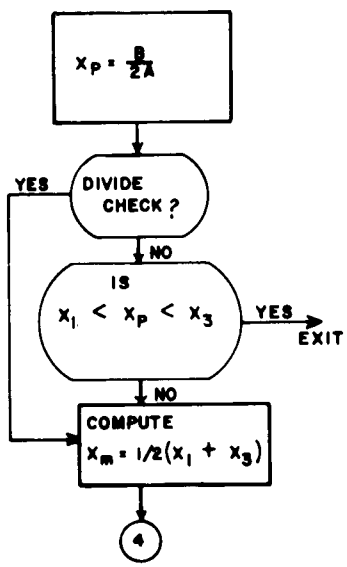
① COMPUTE $X_P$

$$X_P = \frac{B}{2A}$$

WHERE

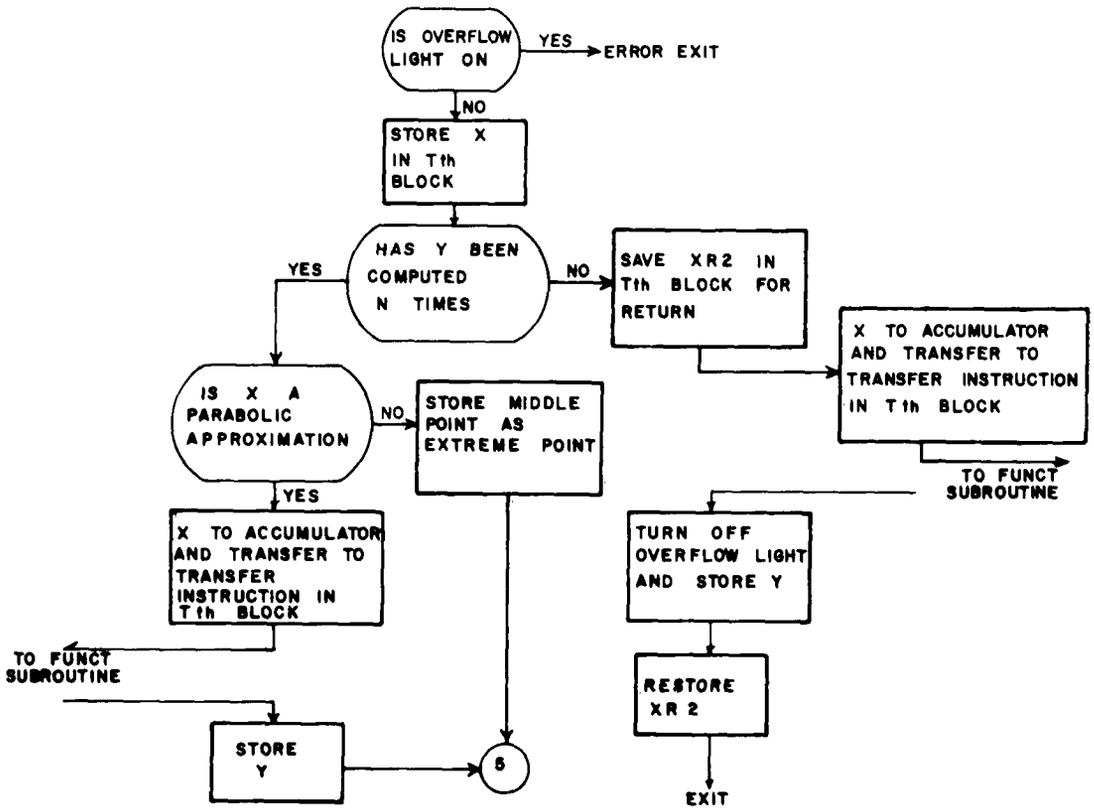
$$A = Y_1 (X_2 - X_3) + Y_2 (X_3 - X_1) + Y_3 (X_1 - X_2)$$

$$B = Y_1 (X_2^2 - X_3^2) + Y_2 (X_3^2 - X_1^2) + Y_3 (X_1^2 - X_2^2)$$



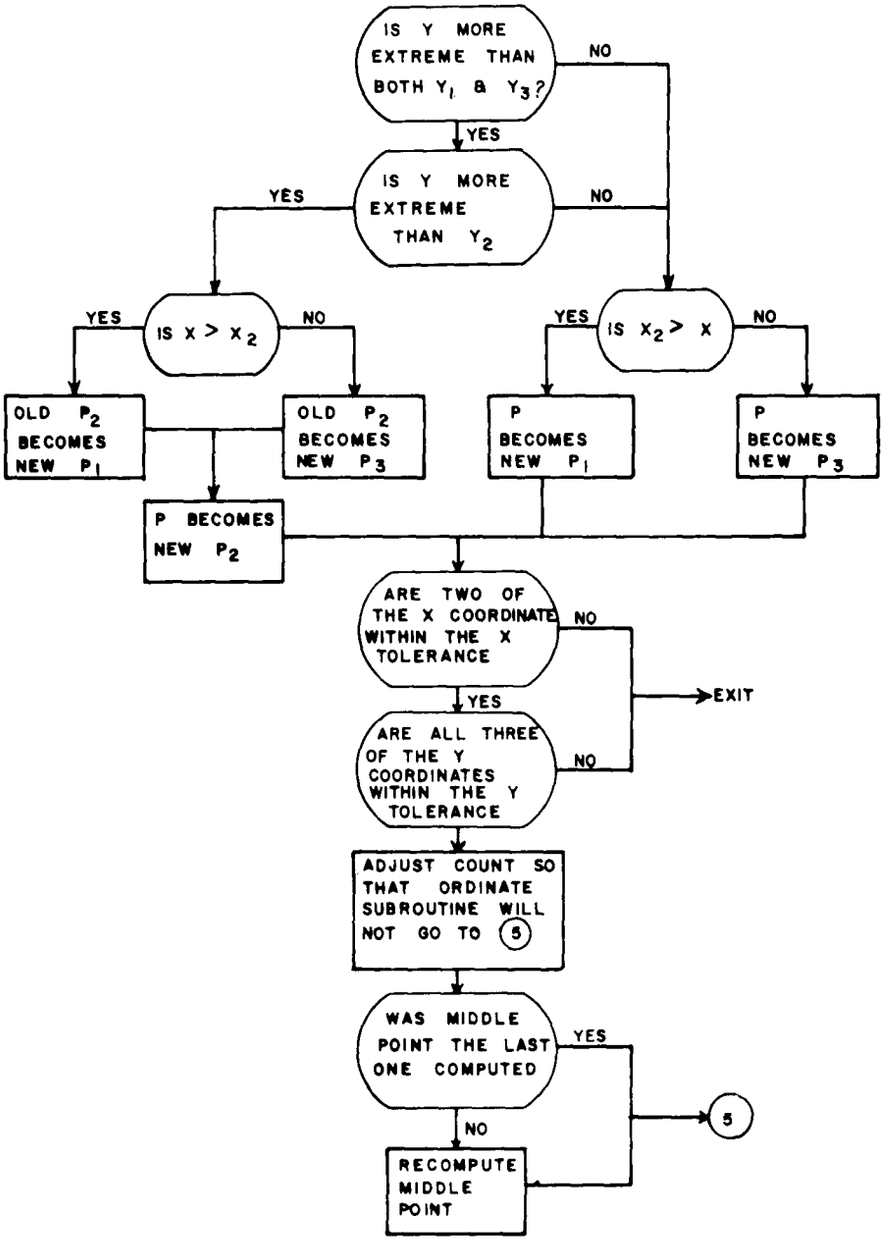
## APPENDIX D (Continued)
## EXTRME CHART 3

② COMPUTE ORDINATE

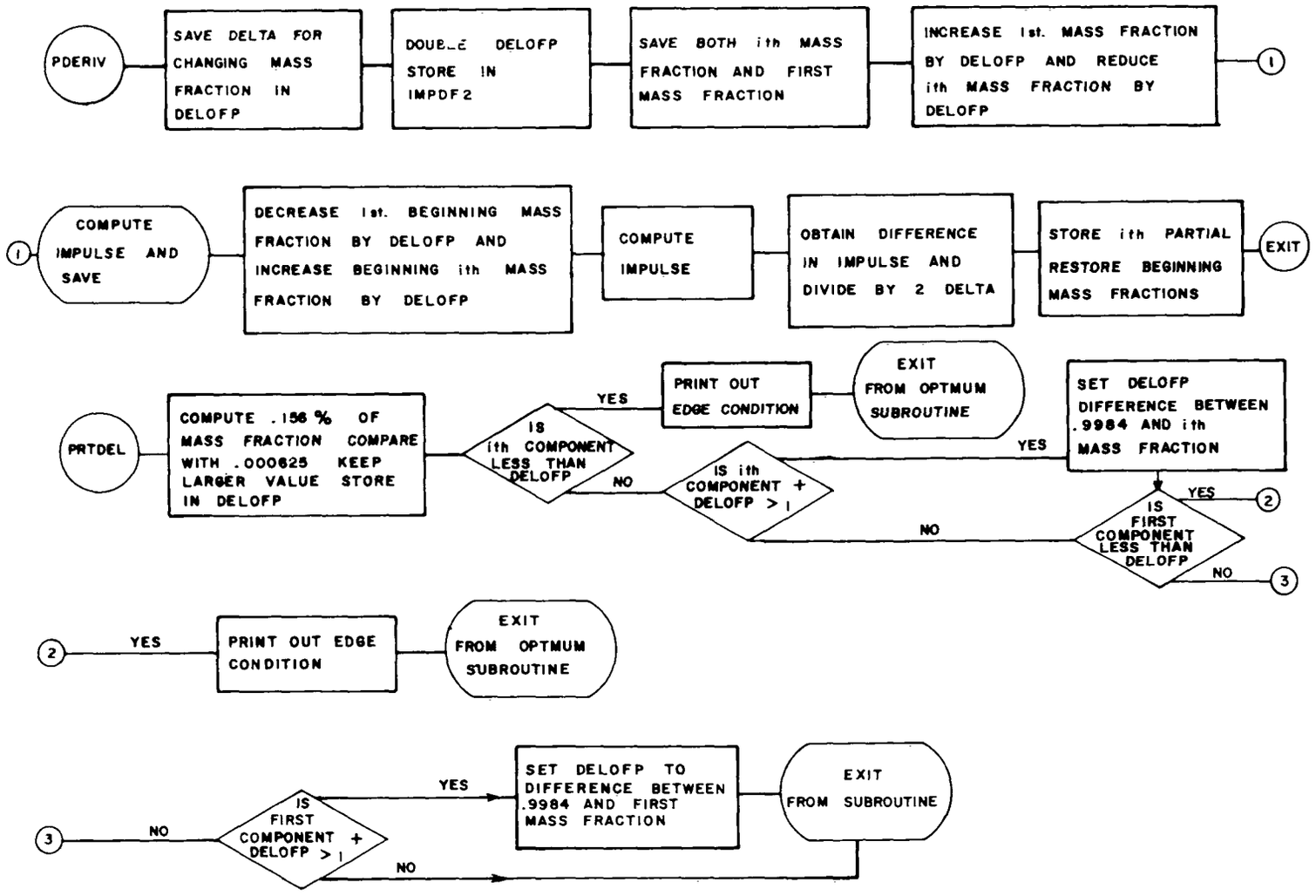


## APPENDIX D (Continued)
## EXTRME CHART 4

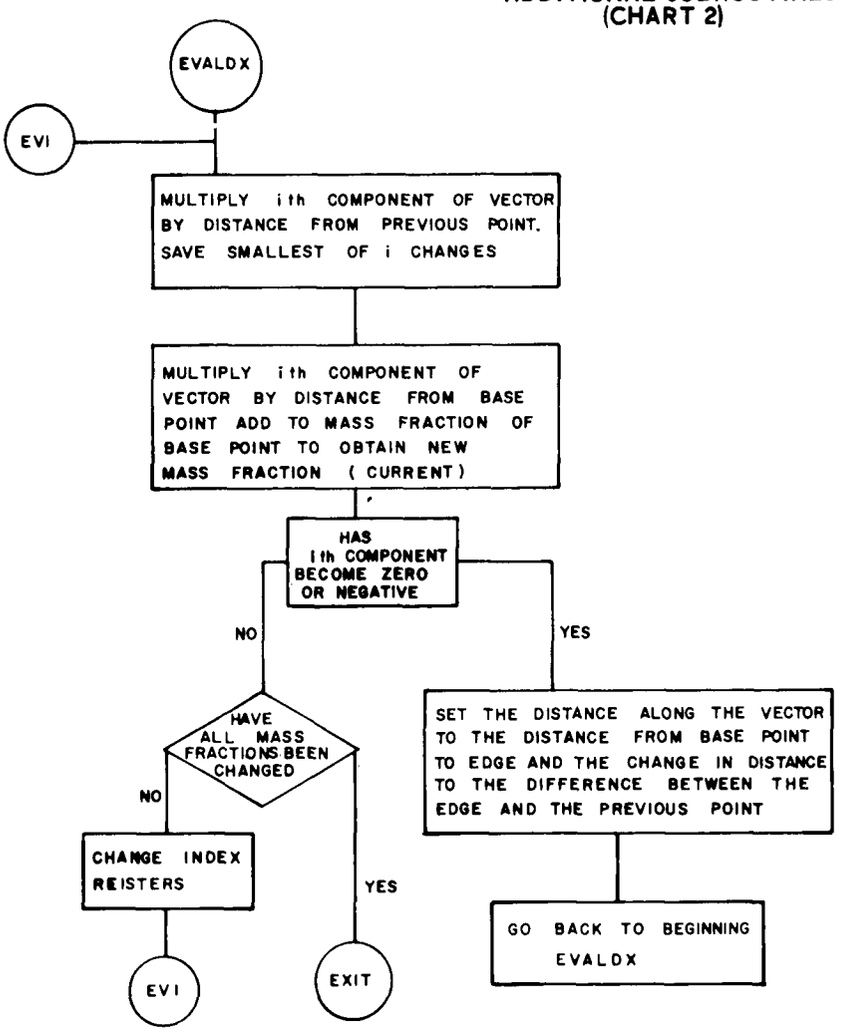
③ GET NEW SET OF POINTS



# APPENDIX E
# ADDITIONAL SUBROUTINES
# (CHART 1)



## APPENDIX E (Continued)
## ADDITIONAL SUBROUTINES
## (CHART 2)



# APPENDIX F

## OPERATING INSTRUCTIONS FOR USE OF OPTIMIZATION SUBROUTINE

The propellant optimization routine is an open subroutine. Several parameters and a function subroutine must be supplied by the user. The parameters and their symbolic names are:

(1) The number of propellant ingredients; KPROP

(2) The mass fractions (or weight percents) of the ingredients, their chemical formula and heat of formation at 298°K; block starting at PROPMF (up to 10 cells)

(3) The gaseous product species mass fractions; block starting at XG (up to 100 cells)

(4) The condensed product species mass fractions; block starting at XC (up to 20 cells)

The function subroutine to be supplied is assumed to have the symbolic entry IMPULS (index register 4 is set upon entry) and should return to the optimization routine by TRA 1, 4 with the function value in the accumulator.

The specification of the reactant mass fractions allows the user to specify the point on the impulse surface from which the optimization routine begins. The product species mass fractions are used to permit the optimization routine to provide the IMPULS subroutine with good estimates in order to minimize the computation time required to calculate specific impulse. The use of the latter capability is not required.

With the completion of initialization, control is transferred to the optimization routine with TSX OPTMUM, 4. Specific impulse is calculated for the original fuel composition and that is regarded as the first base point. The derivatives of impulse with respect to changes in fuel composition are determined numerically. 2n-2 impulse calculations (n = the number of reactant species) are required for derivative calculations at a base point. These derivatives are used to define the line along which a maximum impulse is sought. If the "t" iteration is used (only after at least two base points are established) the line is defined as a linear combination of

the lines established by the numerical derivatives at two successive base points. After the maximum impulse along the line (or among the maxima of several lines if the t iteration is used) is found, the point at which it occurs is used as the next base point and the computation proceeds from there as it did from the previous base point. Convergence is identified in one of two ways:

(1) The impulse values at three successive base points are within an interval of one second.

(2) The use of the "t" iteration offers no improvement to the value of specific impulse at the current base point.

If in the search for maximum specific impulse a reactant species concentration tends to zero, an attempt is made to allow the iteration to proceed with only a small amount of that species present. If the attempted elimination of the species is persistent the computation is terminated

After convergence or termination of the optimization, control is transferred to the instruction following the transfer to the optimization subroutine.